



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 5, May 2023

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

An Overview of Techniques for Vehicle Detection and Counting

Nikhil Pachpawa, Mayur Shingane, Vishal Patil, Nilesh Misal, Prof. Kavita Patil

Department Information Technology, Zeal College of Engineering and Research, SPPU, Narhe, Pune, India

ABSTRACT: One of the most crucial jobs in many traffic circumstances is the detection and classification of cars. The task of vehicle detection and classification in this paper is carried out using a Deep Learning system. Vehicle Detection is the first step described for tracking and counting vehicles. The Mask RCNN deep learning model is used to identify various vehicle kinds. The vehicles are then tracked and counted using the Hungarian algorithm and Kalman filter. To match an object from one frame to another, a Hungarian algorithm is utilised. The Kalman filter algorithm is then used to anticipate the present state and update the state depending on the measurements of the bounding boxes after the relationship between detection and the tracker has been established. The experimental findings indicate that vehicle tracking via detection enhances the results of vehicle tracking and counting. While the vehicle tracking and counting algorithm has been created and tested using AAU-Rainsnow dataset, the vehicle detection and classification algorithm is tested using the IITM-Hetra dataset and AAU-Rainsnow dataset

KEYWORDS: Vehicle Detection, Deep Learning, Convolutional Neural Network

I. INTRODUCTION

Traffic cameras play a crucial role for various purposes such as for estimating queue length, analyzing traffic speed, traffic congestion detection and it can also be used for detection and classification of vehicles. Detection and classification of vehicles is a difficult task when vehicles are occluded that we commonly observed in the heterogeneous traffic or Intelligent traffic management and monitoring scenarios[1]. Vehicles are available in different types and sizes in traffic datasets. Our aim is to focus on vehicle detection and classification from the images and frames acquired from the traffic video dataset. Vehicle Detection is used to determine whether the vehicles are present inside the image or not and it returns its spatial location of the detected vehicles and classify it into their types. To reduce traffic on the road, the important thing is to count the number of vehicles that are present on the road, so this can be easily achieved by using vehicle counting and tracking. Vehicle tracking is used to track multiple vehicles and determine the current position of the vehicles. And it returns the bounding box for each detected object. Then the Hungarian algorithm is used to determine whether the object detected i The Deep Learning method Convolutional neural network has been widely used for the vehicle detection task. Then the Region base Convolutional neural network comes after the CNN, to extract features it used CNN and then used SVM for the classification. Fast RCNN comes into a trend because RCNN takes too much time for training [2]. The selective search method is used in Fast RCNN to find a region of interest which is a time-consuming process [2]. Then after Faster RCNN [3] comes, it uses Region Proposal Network to find the region of interest. The Mask RCNN [4] is an extension of Faster RCNN in which it returns object mask in addition to bounding box and score. So, in this research, the deep learning method Mask RCNN is used to detect each type of vehicles and it classifies the detected vehicles into their class type, which means whether the particular vehicle is a car, bus, truck, bicycle, etc. The very basic step for vehicle tracking and counting is vehicle detection. To perform vehicle detection Mask RCNN is used a n the previous frame is the same as one of the objects in the current frame or not. This algorithm does the association between detection and tracked objects list, and it returns Hungarian matrix as an output, Hungarian matrix is nothing but a match found between detection to tracker list. Once the matches are found, we can easily determine the unmatched tracker and unmatched detection. After that, the Kaman filter algorithm is applied to predict the position of the object. The pre-trained models are not trained for auto class and to detect auto class also, we need to train using a huge dataset and training using the huge dataset it takes too much amount of time. So, Transfer Learning is performed using a pre-trained model to detect and classify auto correctly

II. LITERATURE REVIEW

In the last few years, researchers have shown a great interest in vehicle detection, classification, and also in the area of tracking and counting. Many researchers used deep learning algorithms for vehicle detection and classification task.

This chapter briefly discusses the related research work carried out by different researchers in the area of vehicle detection, classification, counting, and tracking.

D. Mittal et al. (2018) proposed Faster RCNN for vehicle detection and classification. A Large amount of dataset is needed to train a Deep Learning Network so, they have augmented existing large scales datasets and uses Faster RCNN for vehicle detection and classification purpose. They have performed testing using four different approaches: first, they applied Faster RCNN pre-trained model, performed fine-tuning of a pre-trained model with their collected dataset, train the model from the scratch using their dataset, and then they trained model using collected dataset and existing dataset. However, the collected dataset is quite different from the existing dataset. So, they have augmented the Pascal VOC dataset[5] with the IITM-HeTra[6] dataset. so using this way improves the performance of vehicle detection.[1]

B. Hicham et al. (2018) presented two methods: Firstly they have implemented the Data Augmentation technique to reduce the imbalanced dataset problem and secondly they apply the CNN model to perform vehicle detection and classification. The structure of CNN comprises of series of convolutive layers and fully connected layers. The convolutive layer used to perform feature extraction and fully connected Artificial Neural Networks (ANN) layers used to classify the objects. Precision, Recall, and Accuracy metrics used to measure the performance of the system.[7]

M. V. et al. (2019) proposed Deep Learning Framework R-CNN for the classification of different types of vehicles. Box filter smoothening technique is implemented to reduce the noise in the image. The background information is removed from the frame to detect the interested region.[8]

K. Shi et al. (2017) presented Fast RCNN for the purpose of vehicle detection. An author has performed efficient vehicle detection using incremental learning and pre-processing approach to optimize the training process, training parameters are adjusted during the training process in order to achieve the best state. The vehicle detection model comprises of the two stages: 1) training stage and 2) testing stage. Inside the training stage, they re- trained the initial parameters of the CNN after pretraining on ImageNet. The testing stage is used to test the input samples. CaffeNet, VGG_Cnn_M_1024, and VGG-16 networks are used in the pretraining process. Fast RCNN is initialized with the CaffeNet but the ROI pooling layer is used in place of the Last pooling layer, using two parallel layers instead of a fully connected layer and softmax layer, and it takes two sets of data as an input. The new dataset is formed for incremental learning by adding the BUU-T2Y dataset to the KITTI dataset[9] and using this it improves the vehicle detection performance.[10]

C. N. Aishwarya et al. (2018) proposed the YOLO method for object detection purposes. To achieve better prediction accuracy they trained a Neural network named Inception V3 with the dataset that comprises of two different types of images, both static images, and dynamic images. After that, they used image processing methods such as Sobel kernel, and thresholding to identify license plate and use classification algorithm KNN to determine the characters. Accuracy of Tiny YOLO is compared with the YOLO. As compared to YOLO, Tiny YOLO is less accurate and it classifies only fewer objects.[11]

B. Benjdira et al. (2018) have examined the performance of Faster RCNN and YOLOv3 for the purpose of car detection. They have trained a dataset using both Faster RCNN and YOLOv3. After training, try to evaluate both algorithms using the F1 score, precision, recall, processing, and quality time. Both of the algorithms have high precision, it means both will able to recognize the car, but in case of a recall, YOLOv3 is better than the Faster RCNN.[12]

G. Prabhakar et al. (2017) proposed Faster RCNN to detect on-road objects. The ZF Net pre-trained model is fine-tuned with the PASCAL VOC 2012 dataset to detect only 20 objects. Training is available in two ways: Approximate Joint Training, it trains both the RPN and Fast R-CNN at the same time. While in the Alternate Training, first train the RPN and then train Fast R-CNN using generated region proposals. They have performed training using Approximate Joint Training. They have tested results on a variety of datasets under different climate conditions. The performance of vehicle detection is measured using Mean Average Precision (mAP). This method will able to detect all the on-road objects but it sometimes fails to detect auto class.[13]

M. C. Olgun et al. (2018) presents Faster RCNN and Haar Cascade Classifier. They performed training using Faster RCNN and used feature extractor Resnet Inception V2 to detect the vehicles. To detect traffic light and stop sign they used Haar Cascade Classifier. CNN based on NVIDIA's PilotNet is used for the purpose of lane tracking and they trained the model using CNN and applied data augmentation to support a variety of lanes.[14]

S. Srilekha et al. (2015) have presented vehicle detection, tracking, and counting using a Kalman filter. Moving object detection using a background subtraction technique is difficult. So they used the Kalman filter algorithm and using this technique they achieve better object detection accuracy compared to the background subtraction.[15]

P. K. Bhaskar and S.-P. Yong (2014) used Gaussian Mixer Model(GMM) and blob detection for vehicle data recognition and tracking purpose. Some morphological operations have been applied to detect moving objects and remove noise. Gaussian Mixer Model is used to separate the foreground and background of the image. The Blob detection used to determine the object movement in the frame.[16]

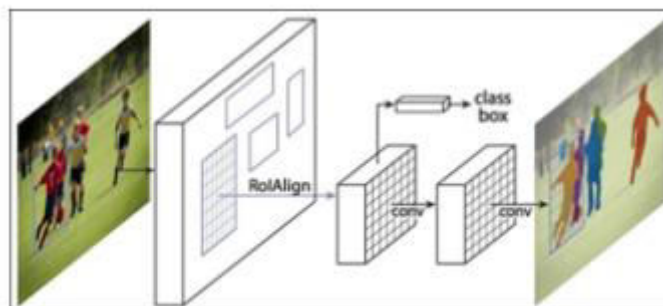
M. Djalalov et al. (2010) perform object detection using Median Filtering and Blob extraction and track the vehicles using Kalman filter. First, they apply the median filter and it results in a background image. After that, apply background subtraction to determine the moving objects for each frame. They combined edge pixels into the objects by performing morphological closing operations. Then the location of vehicles is tracked using the Kalman filter.[17]

III. PROPOSED SYSTEM

Mask RCNN-Based Vehicle Detection and Classification

The most crucial task in detecting all forms of cars is vehicle detection. First, the video frames must be extracted before any vehicles may be detected from the video sequence. Apply the Deep Learning Framework Mask RCNN to each frame after removing the frames from the movie in order to identify every type of vehicle in the frame and classify the detected vehicles according to their types, such as a car, bus, truck, bicycle, etc.

Figure 3.1 Mask RCNN



Vehicle Counting from Image

Figure 4.5 shows the results of vehicle detection and classification, and Figure 4.6 shows the total number of vehicles that are detected inside the figure 4.5, there are 6 vehicles detected in the below image and all detected vehicles are of type class car.

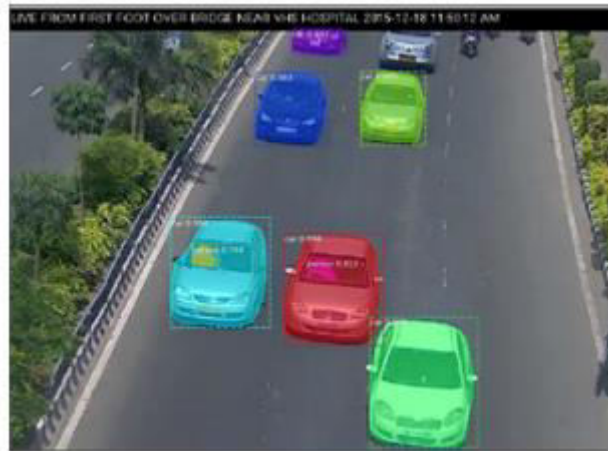


Figure 4.5 Vehicle Detection and Classification from Image

IV. METHODOLOGY

Dataset Preparation:

Gather a large dataset of annotated images containing vehicles and non-vehicle objects.

Annotate the dataset with bounding boxes around vehicles to serve as ground truth for training the CNN model.

Preprocessing:

Resize the images to a fixed size suitable for input to the CNN model.

Normalize the pixel values to a common scale (e.g., [0, 1]).

Model Architecture:

Design a CNN architecture suitable for object detection tasks. Common architectures include Faster R-CNN, YOLO (You Only Look Once), or SSD (Single Shot MultiBox Detector).

Initialize the CNN model with pre-trained weights on a large-scale dataset like ImageNet to leverage learned features.

Training:

Split the annotated dataset into training and validation sets.

Feed the training images through the CNN model and optimize the model's parameters using a suitable loss function (e.g., cross-entropy loss).

Adjust the model's hyperparameters, such as learning rate and batch size, to optimize performance.

Regularize the model to prevent overfitting by techniques like dropout or weight decay.

Iterate the training process until the model converges and achieves satisfactory performance on the validation set.

Vehicle Detection:

Apply the trained CNN model to new images for vehicle detection.

Pass the image through the CNN model to generate object proposals or bounding box predictions.

Utilize non-maximum suppression to eliminate redundant and overlapping bounding boxes.

Threshold the remaining bounding boxes based on their confidence scores to filter out false positives.



Vehicle Counting:

Implement a tracking algorithm to assign unique IDs to each detected vehicle and track them across consecutive frames.

Use techniques like Kalman filtering or Hungarian algorithm for tracking vehicles and maintaining track consistency.

Count the number of unique IDs assigned to vehicles as the vehicle count for a given image or video sequence.

Postprocessing and Evaluation:

Perform postprocessing steps, such as removing spurious detections or refining bounding box coordinates.

Evaluate the performance of the vehicle detection and counting system using appropriate metrics such as precision, recall, and F1 score.

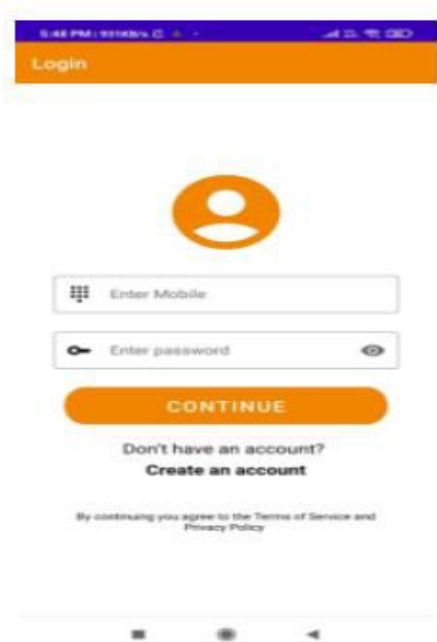


Fig. No.1: Log in page

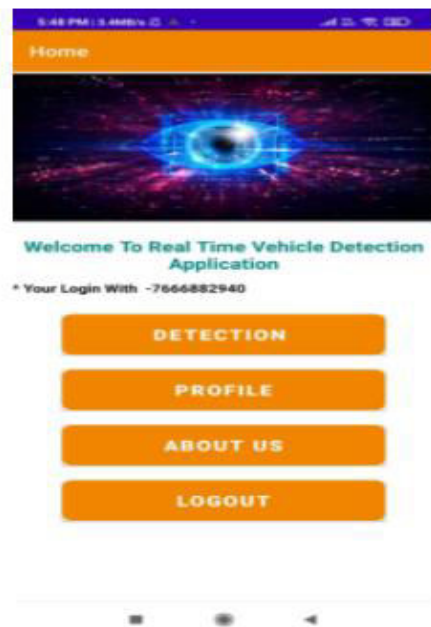


Fig. No.2 : Home page



|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205400 |

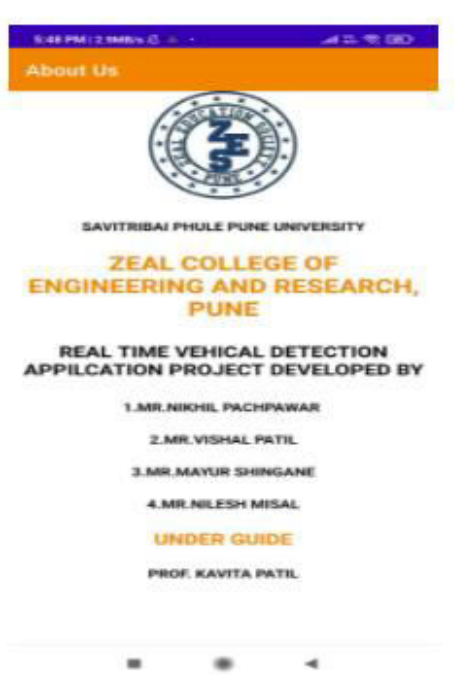


Fig. No.3:About Us

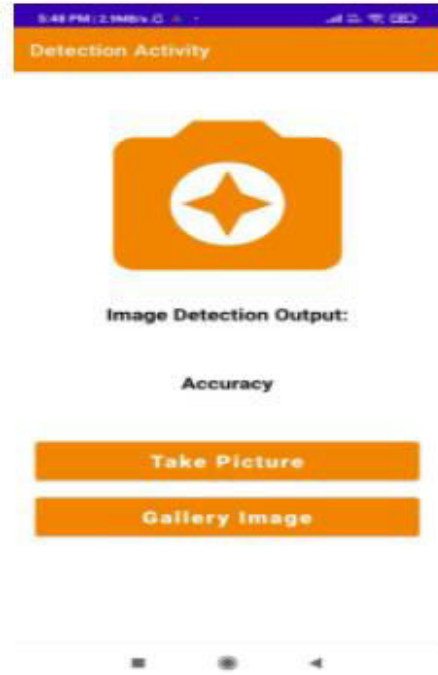


Fig. No.4:Detection Activity



Fig. NO.5:Image Count

V. CONCLUSION AND FUTURE WORK

The detection and classification of vehicles are performed using Mask RCNN deep learning algorithm. The different types of vehicles are easily detected and classified using the Mask RCNN algorithm. The Vehicle Tracking and Counting task are performed using the Hungarian algorithm and Kalman Filter algorithm. The Hungarian algorithm is

used to determine whether the vehicle detected in the previous frame is the same as the vehicle detected in the current frame or not. The Kalman Filter is used to predict and update the state. The performance of the vehicle tracking and counting is tested using IITM-Hetra and AAU-Rainsnow dataset and found that it gives better results of tracking and counting.

Transfer Learning is also performed to detect the auto class. To improve the result of transfer learning, some experiments are performed by changing the batch size.

In the future, try to improve the model so it will be able to track all the vehicles and count the vehicles accurately.

REFERENCES

1. D. Mittal, A. Reddy, G. Ramadurai, K. Mitra, and B. Ravindran, "Training a deep learning architecture for vehicle detection using limited heterogeneous traffic data," in *2018 10th International Conference on Communication Systems Networks (COMSNETS)*, Jan. 2018, pp. 589–294, doi: 10.1109/COMSNETS.2018.8328279.
2. "Introduction to Object Detection Algorithms," *Analytics Vidhya*, Oct. 11, 2018. <https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/> (accessed Jul. 11, 2020).
3. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
4. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Oct. 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.
5. "The PASCAL Visual Object Classes Challenge 2012 (VOC2012)." <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/> (accessed Jun. 01, 2020).
6. "IITM-HeTra." <https://kaggle.com/deepak242424/iitmhetra> (accessed Jun. 01, 2020).
7. B. Hicham, A. Ahmed, and M. Mohammed, "Vehicle Type Classification Using Convolutional Neural Network," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, Marrakech, Oct. 2018, pp. 313–316, doi: 10.1109/CIST.2018.8596500.
8. M. V. V. R., and N. A., "A Deep Learning RCNN Approach for Vehicle Recognition in Traffic Surveillance System," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, Apr. 2019, pp. 0157–0160, doi: 10.1109/ICCSP.2019.8698018.
9. "The KITTI Vision Benchmark Suite." http://www.cvlibs.net/datasets/kitti/raw_data.php (accessed Jun. 01, 2020).
10. K. Shi, H. Bao, and N. Ma, "Forward Vehicle Detection Based on Incremental Learning and Fast R-CNN," in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, Hong Kong, Dec. 2017, pp. 73–76, doi: 10.1109/CIS.2017.00024.
11. C. N. Aishwarya, R. Mukherjee, and D. K. Mahato, "Multilayer vehicle classification integrated with single frame optimized object detection framework using CNN based deep learning architecture," in *2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, Mar. 2018, pp. 1–6, doi: 10.1109/CONECCT.2018.8482366.
12. B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3," *ArXiv181210968 Cs*, Dec. 2018, Accessed: Jun. 01, 2020. [Online]. Available: <http://arxiv.org/abs/1812.10968>.
13. G. Prabhakar, B. Kailath, S. Natarajan, and R. Kumar, "Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving," in *2017 IEEE Region 10 Symposium (TENSYMP)*, Cochin, India, Jul. 2017, pp. 1–6, doi: 10.1109/TENCONSpring.2017.8069972.
14. M. C. Olgun, Z. Baytar, K. M. Akpolat, and O. Koray Sahingoz, "Autonomous Vehicle Control for Lane and Vehicle Tracking by Using Deep Learning via Vision," in *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, Istanbul, Turkey, Oct. 2018, pp. 1–7, doi: 10.1109/CEIT.2018.8751764.



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.423



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com



www.ijirset.com

Scan to save the contact details